UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,060	07/06/2001	Travis J. Muhlestein	MSFT115921	7821

26389 7590 10/11/2007
CHRISTENSEN, O'CONNOR, JOHNSON, KINDNESS, PLLC
1420 FIFTH AVENUE
SUITE 2800
SEATTLE, WA 98101-2347

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
10/11/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

mn

Office Action Summary	**Application No.** 09/900,060	**Applicant(s)** MUHLESTEIN ET AL.	
	Examiner Tuan A. Vu	**Art Unit** 2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 June 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1 and 3-17* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1 and 3-17* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 6/04/07.

As indicated in Applicant's response, claims 1, 3, 8, 17 have been amended, and claim 2 previously canceled. Claims 1, 3-17 are pending in the office action.

Claim Objections

2. Claim 17 is objected to because of the following informalities: there appears to be a extraneous 'for performing' after 'to carry out'.

Appropriate typographical correction is required.

Claim Rejections - 35 USC § 101

3. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

4. Claims 7 and 16 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

> The Federal Circuit has recently applied the practical application test in determining whether the claimed subject matter is statutory under 35 U.S.C. § 101. The practical application test requires that a " useful, concrete, and tangible result" be accomplished. An "abstract idea" when practically applied is eligible for a patent. As a consequence, an invention, which is eligible for patenting under 35 U.S.C. § 101, is in the "useful arts" when it is a machine, manufacture, process or composition of matter, which produces a concrete, tangible, and useful result. The test for practical application is thus to determine whether the claimed invention produces a "useful, concrete and tangible result".

The current focus of the Patent Office in regard to statutory inventions under 35 U.S.C. § 101 for method claims and claims that recite a judicial exception (software) is that the claimed invention recite a practical application. Practical application can be provided by a physical transformation or a useful, concrete and tangible result. The following link on the World Wide Web is for the United States Patent And Trademark Office (USPTO) policy on 35 U.S.C. §101.

<http://www.uspto.gov/web/offices/pac/dapp/opla/preognotice/guidelines101_20051026.pdf>

Specifically, claims 7 and 16 recite 'computer-readable medium' to perform the method of claim 1. However, the Disclosure clearly indicates that 'computer-readable media' may comprise 'computer storage media and communication media' (Specifications, pg. 10, 1st para), and Fig. 1 and description of such communication media depicts Internet connectivity between computers (Specifications, pg. 9, 2nd para; pg. 11, 2nd para), e.g. a transmitted signal between modems. The form of media being conveyed as internet transmission media (e.g. via a modem) amounts to non-tangible storage medium hence would not qualify the above 'computer-readable medium' as a *statutory category* (refer to the above 101 Guidelines *pdf* file, pg. 14-16). As a whole, the claim for failing to belong to one of the statutory categories will be rejected for leading to a non-statutory subject matter. Suggested correction for this is that the readable medium has to be claimed 'computer-readable storage medium'.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. Claims 1, 3-17 rejected under 35 U.S.C. 103(a) as being unpatentable over Microsoft Corporation, "Microsoft Windows Management Instrumentation Scripting", April 1999, pp. 1-15 (hereinafter MSWMI), further in view of Admitted Prior Art (APA – see BACKGROUND of application).

As per claim 1, MSWMI discloses a computer-implemented method for providing access to instrumentation data from within a managed code runtime environment, the method comprising

providing an application (e.g. WMI technology – Introduction) from in a runtime-aware programming language (e.g. Introduction: *enterprise environment, model* - pg 1; Object, Information Model: pg. 5-11), the application being suitable for execution by a runtime engine in a managed runtime environment (Note: application of a enterprise application with modeling and interface or extension APIs reads on application with a runtime-aware programming language);

executing the application in a managed runtime environment having a runtime engine, wherein there is a defined contract of operation between the executing application and the runtime engine to delegate certain application tasks to the runtime engine that enable the runtime engine to service requests (e.g. **Windows Management Instrumentation Technology**: *Access to monitor, command, control any entity…underlying mechanism, API … Interoperability …providing and accessing management …extend the information …connect one or more sources of management information …capture instrumentation, detailed queries* --pg. 1, bottom to pg. 2 , top) from the executing application at runtime;

including requests for instrumentation data representing management information about other applications and devices available outside the runtime environment (e.g. *to capture instrumentation data from device drivers kernel* ..- pg. 2, 5th bullet-top; *Performance Monitor Provider* – pg. 4, 4th bullet; **WMI Architecture Overview**: *using WMI APIs … providers supply … CIM object Manager with data from managed objects, handle requests* ; *interface between* *management applications and data providers ... common programming interface to Windows Management Instrumentation,*– pg.3, middle; Fig. 1, pg. 4; *WMI Providers data that is not available from the CIMOM ... forward to WMI Provider data and event notifications for managed objects* – top pg. 4; **Advantages of Using WMI Scripting**: *custom providers can ... cover vendor specific instrumentation (for system, applications, devices...), Extensible Providers instrumentation* – 3rd bullet, pg. 5);

receiving a request at the runtime engine from the executing application for instrumentation data available outside said managed code runtime environment the request including

a path of an instrumentation data object (e.g. *SWbemObjectPath* – pg. 6, Features: Object Creation; SWbemObjectPath – bottom, pg. 7) for accessing the instrumentation data (e.g. pg. 2, 5th bullet-top; pg.3, middle; Fig. 1, pg. 4; top pg. 4),

options used to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf ...* pg. 7, middle; *GetObjectText_, SpawnInstance_*, pg. 9, middle) the instrumentation data object, and

an identification of a parent (e.g. *ParentNameSpace*, pg. 8, 3rd bullet)of the instrumentation data object;

transmitting a corresponding request for said instrumentation data to an instrumentation data source external to said managed code runtime environment, receiving a response to said corresponding request from said instrumentation data source (e.g. *to capture instrumentation data from device drivers kernel ..*- pg. 2-top, 5th bullet; **WMI Architecture Overview**: *using WMI APIs ... providers supply ... CIM object Manager with data from managed objects, handle* *requests* ; *interface between management applications and data providers ... common programming interface to Windows Management Instrumentation,*– pg.3, middle; Fig. 1, pg. 4; *WMI Providers data that is not available from the CIMOM ... forward to WMI Provider data and event notifications for managed objects* – top pg. 4);

converting said response to a format that is compatible with said managed code runtime environment (**Windows Management Instrumentation Technology**: *supports the syntax of CIM, MOF, common programming interface, scripting support* - pg. 1, bottom – Note: WMI environment working in conjunction with providers via scripting, and API for retrieval of remote objects, while supporting syntax of all interfaces reads on converting to syntax compatible for the WMI);

responding to said request for instrumentation data with said converted response (Note: request for data using API and collecting data into a compatible form for the modeling/instrumentation application reads on responding to request for such instrumentation data).

MSWMI does not explicitly discloses that the application for the runtime-aware language is written in a intermediate language, nor does MSWMI disclose that the runtime engine to execute said application is configured to execute such intermediate language. The use of WMI (Microsoft WMI or MSWMI) in application environment known as .NET platform has been well-established at the time the invention was made as set forth in APA (see BACKGROUND of Application: pg. 3, bottom para; pg. 4, top two para), according to which Microsoft .NET platform utilizes Microsoft WMI to effect the interface necessary to retrieve instrumentation data which is taught by MSWMI to the .NET platform, wherein .NET application is compiled as intermediate code (IL) so that the IL is admittedly being run using by a Microsoft .NET runtime engine (APA, see BACKGROUND: pg. 2, 3rd para). Based on MSWMI being also a Microsoft product used in retrieving instrumentation data for Microsoft runtime application, it would have been obvious for one of ordinary skill in the art at the time the invention was made to utilize the WMI (as by MSWMI) so that it supports as interface to a application written in IL and executed by a .NET runtime engine (as by APA) because according the Microsoft and APA, .NET applications programs are platform independent designed to communicate with many other sources, and since MSWMI is also product of Microsoft running as interface in its own form in tandem with the Microsoft .NET environment (see APA, pg. 3-4) for rendering a variety of services to retrieve such multi-source data for the managed code of .NET (see APA), using the WMI into support a Microsoft .NET application as set forth by APA would be the very purpose of WMI (see APA: BACKGROUND, pg. 4) in light of .NET methodology's endeavor to obtain instrumentation data as purported by MSWMI.

As per claim 3, MSWMI discloses converting instrumentation data object to a management object that is compatible with said runtime environment (see claim 1; *Using WMI technology .. create ...applications that implement ... features such as displaying system information, generating ... inventory resources...processing events* – pg. 3, Management Applications, bottom – Note: integrating data from request via API calls in order to integrate them for display in application via processing therein <u>reads on</u> converting requested data in runtime compatible form).

As per claim 4, MSWMI discloses wherein said management object encapsulates properties of the instrumentation data object (e.g. Standard inheritable methods – pg. 3, top, 2nd bullet; **Features**: *Monikers, for encapsulating the location* - pg 6, middle) accessible through said instrumentation data source, including

properties representing the path (e.g. Features: Object Creation, pg. 6; SWbemObjectPath – bottom, pg. 7) of the instrumentation data object for accessing the instrumentation data,

the options used to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf*... pg. 7, middle) the instrumentation data object and

the identification of the parent (e.g. *ParentNameSpace*, pg. 8, 3rd bullet) of the instrumentation data object.

As per claims 5-6, MSWMI discloses wherein said response comprises an indication that an operation was unsuccessful and wherein converting said response to said format comprises generating a management exception object; said indication that an operation was successful comprises error codes (e.g. **Advantage of Using WMI Scripting**: 4th bullet: *built-in features ... exception* --pg. 5, middle; **Features**: *Error Handling* - pg 6, middle; **Asynchronous example:** *hResult, ErrorObject* – pg. 14, 2nd para; **SwbemLastError object**: *read-once semantics... cleared after reading* – pg. 9, bottom).

As per claim 7, MSWMI discloses a computer-readable medium comprising instructions which, when executed by a computer, cause the computer to perform the method of any one of claims 1 and 3-6 (e.g. Note: a computer system capable of supporting script, encapsulating of objects, API calls, binding object-oriented instances to a model, and display of instrumentation data or event processing as in claims 1, 3-6 reads on inherent computer readable medium for storing such software capabilities).

As per claim 8, MSWMI discloses a computer-controlled apparatus comprising a processing unit and a system memory, and wherein the apparatus further comprises a managed code runtime environment and is configured to carry out the method of any one of claims 1 and 3-6 (see claim 7).

As per claim 9, MSWMI discloses a computer-implemented method for accessing instrumentation data from within a runtime environment, wherein the runtime environment provides a runtime engine that executes an application compiled in a runtime-aware language (e.g. Introduction: *enterprise environment, model* - pg 1; Object, Information Model: pg. 5-11-- Note: application of a enterprise application with modeling and interface or extension APIs reads on application with a runtime-aware programming language), the method comprising:

receiving a request from the application for instrumentation data representing management information about other applications and devices available outside the runtime environment (*to capture instrumentation data from device drivers kernel* ..- pg. 2-top, 5th bullet; **WMI Architecture Overview**: *using WMI APIs ... providers supply ... CIM object Manager with data from managed objects, handle requests* ; *interface between management applications and data providers ... common programming interface to Windows Management Instrumentation,*– pg.3, middle; Fig. 1, pg. 4; *WMI Providers data that is not available from the CIMOM ... forward to WMI Provider data and event notifications for managed objects* – top pg. 4,

the request comprising a path of an instrumentation data object for accessing said instrumentation data (e.g. Features: Object Creation, pg. 6; SWbemObjectPath – bottom, pg. 7), options used to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf* ... pg. 7, middle; *GetObjectText_, SpawnInstance_*, pg. 9, middle) the instrumentation data objects and a namespace (e.g. **SWbemServices object**: *object ...connection to a namespace* – pg. 7, middle; *ParentNameSpace*, pg. 8, 3rd bullet) of the instrumentation data object;

in response to said request, querying for said instrumentation data, using the path to said instrumentation data object for accessing said instrumentation data; determining whether said instrumentation data was successfully returned (**WMI Scripts Usage**: *Method Execution, Queries, remote Access*, pg. 11; **Asynchronous example:** *hResult, ErrorObject* – pg. 14, 2nd para – Note: scripting with path parameters reads on using path to incorporate in the query effected via API calls); and

in response to determining that said instrumentation data was successfully returned, constructing said management object in the runtime environment and populating said management object (e.g. CIM Object Collection-*SwbemObjectSet*, pg 11 – Note: object set after collecting of data from remote access reads on populating CIM model; **Features**:Object Creation, Collections, Direct Access, pg. 6; SwbemEventSource Object, SwbemNamedValueSet collection, SwbemObject) with said instrumentation data.

MSWMI does not explicitly disclose that the application for the runtime-aware language is written in an intermediate language. But this limitation has been addressed in claim 1 above.

As per claim 10, MSWMI discloses wherein constructing said management object in the runtime environment and populating said management object with said instrumentation data includes binding an instance of a management object class (e.g. **Features**: *Monikers* – pg. 6, middle) to said instrumentation data object for accessing said instrumentation data source.

As per claim 11, MSWMI discloses constructing a management scope object identifying the namespace (**SWbemServices object**: *object ...connection to a namespace* – pg. 7, middle; *ParentNameSpace*, pg. 8, 3rd bullet) associated with said instrumentation data object for accessing said instrumentation data.

As per claims 12-13, MSWMI discloses constructing a management path object identifying the path (Features: Object Creation, pg. 6; SWbemObjectPath – bottom, pg. 7), and specifying the options to retrieve (e.g. *SWbemServices Object: Get, Delete, InstancesOf, ExecQuery, AssociatorsOf*... pg. 7, middle; *GetObjectText_, SpawnInstance_*, pg. 9, middle) said instrumentation data object for accessing said instrumentation data.

As per claim 14, MSWMI discloses throwing a management exception object (**Advantage of Using WMI Scripting**: 4th bullet: *built-in features ... exception* --pg. 5, middle; **Features**: *Error Handling* - pg 6, middle; **Asynchronous example:** *hResult, ErrorObject* – pg. 14, 2nd para; **SwbemLastError object**: *read-once semantics... cleared after reading* – pg. 9, bottom) in response to determining that said instrumentation data was not successfully returned.

As per claim 15, MSWMI discloses wherein properties of said management object may be accessed utilizing an indexer (e.g. *SwbemNamedValueSet: ...indexing mechanism* – **SwbemNamedValueSet collection,** pg. 8).

As per claims 16-17, MSWMI discloses a computer-readable medium and computer-controlled apparatus comprising a processing unit and a system memory, and wherein the apparatus further comprises a managed code runtime environment and is configured to carry out [for performing] the method of any one of Claims 9-15 (refer to claims 7-8).

Response to Arguments

7. Applicant's arguments filed 6/04/07 have been fully considered but they are not persuasive. Following are the Examiner's observation in regard thereto.

35 USC § 103 Rejection:

(A) Applicants have submitted that for claim 1, access to platform specific data/functions has been unavailable, as per APA, and that it would be contradictory to develop platform independent environment and incorporate therein specific API as the Office Action suggests(Appl. Rmrks pg. 10, top). The rejection is intended to address whether implementing a API invoked in the runtime of a MSWMI environment in light of APA would have been obvious, and the rationale to combine BACKGROUND teaching and the MSWMI reference has been textually as follows:

Based on MSWMI being also a Microsoft product used in retrieving instrumentation data for Microsoft runtime application, it would have been obvious for one of ordinary skill in the art at the time the invention was made to utilize the WMI (as by MSWMI) so that it supports as interface to a application written in IL and executed by a .NET runtime engine (as by APA) because according the Microsoft and APA, .NET applications programs are platform independent designed to communicate with many other sources, and since MSWMI is also product of Microsoft running as interface in its own form in tandem with the Microsoft .NET environment (see APA, pg. 3-4) for rendering a variety of services to retrieve such multi-source data for the managed code of .NET (see APA), using the WMI into support a Microsoft .NET application as set forth by APA would be the very purpose of WMI (see APA: BACKGROUND, pg. 4) in light of .NET methodology's endeavor to obtain instrumentation data as purported by MSWMI.

If the onset (as set forth in the 103 rejection preamble) is to implement a program interface in a intermediate form as derived from the very nature of the executing (managed code) environment of WMI in view of the .NET context, one of ordinary skill in the art would not unreasonably implement a API so that this API is merely written for a very specific platform, because this would be defeating the very purpose of having intermediate form of code, like bytecodes or IL as suggested by the well-established MSWMI and the portability of Object-Oriented application/programming language upon which said .NET developmental methodology has been founded. There is not a sentence in the above (excerpt of) the 103 rationale that states integrating a platform or architecture-specific API inside the above managed code runtime or .NET Microsoft application. The argument appears to rely on a non-existing statement (i.e. 'incorporate platform specific API') as though this is found from the Office Action in order to attempt to denigrate this very rationale as set forth above. The argument is without foundation and is not persuasive.

(B) Applicants have submitted that the Office Action is suggesting that obviousness is defined rather as a function of where you work than as a function of one of ordinary skill in the art; and this is contrary to rules for determining obviousness (Appl. Rmrks pg. 10, middle). The rationale of obviousness has set up 3-4 substeps of prima facie a rationale; that is, determining what the reference MSWMI does not explicitly teach; what MSWMI (Microsoft tool for instrumentation operable within a highly portable code environment) in light of .NET environment (APA – intercommunication to exchange data/application by Microsoft as in Web-based application data exchange) shares in terms of endeavoring a inter-machine platform independent applicability; establishing a need/motivation for a positive result based on one of ordinary skill in the art when presented with the above evidences of what existed at the time the invention was made; and providing benefit or reasonable level of success from combining MSWMI and what has been suggested from APA based on said motivation or desirability as construed by the ordinary skill in the art. The rejection does not include any remote indication that the reason to combine would be based solely on some high standards belonging to some inventors pertinent to some company, as asserted by Applicants. Applicants fail to justify how the effect of combining the Microsoft .NET Java Application with MSWMI in that the API for instrumenting the WMI application would be directly done in IL or bytecodes (as proffered by the Office Action), would be largely inapposite with the object-oriented code applicability so instrumental to the environments of .NET or WMI. The argument is largely geared off toward a observation that appears to be unrrelated to very grounds of the Office action's rationale.

(C) Applicants have submitted that by combining WMI with .NET as presented by the Office Action, hindsight analysis has been used based on Applicant's Disclosure (Appl. Rmrks bottom pg. 10, top pg. 11). In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971). The rejection has been laid out to include observations according to what is permissible for a standard USC 103 prima facie case, and that has been explained above in section B; all the parts of which observations falling under the level of ordinary skill in the art when faced with the facts or evidence at hands, when APA and the reference are deemed not being part of the Applicant's Disclosure per se. That is, the rationale in so doing takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper hence no hindsight from gleaning the Specifications being applied. The argument is not convincing.

(D) Applicants have submitted that applying contradictory nature of the environment involved and using hindsight by the rationale have the basis of the Office Action for rendering obviousness (Appl. Rmrks pg. 11, middle). This argument has been addressed in the above sections.

(E) Since the issues concerning claims 3-8, 9, 11-17 are based on the arguments for claim 1, these claims also stand rejected; and in all the claims will be rejected as set forth in the Office Action.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on (571)272-3756.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Tuan A Vu
Patent Examiner,
Art Unit 2193
October 10, 2007